



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/670,944	09/25/2003	Pratap Pereira	1014-068US01/JNP-0312	2053
72689 7590 07/24/2008 SHUMAKER & SIEFFERT, P.A 1625 RADIO DRIVE , SUITE 300 WOODBURY, MN 55125			EXAMINER PATEL, DHAIRYA A	
			ART UNIT 2151	PAPER NUMBER
			NOTIFICATION DATE 07/24/2008	DELIVERY MODE ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

pairedocketing@ssiplay.com

Office Action Summary	Application No. 10/670,944	Applicant(s) PEREIRA, PRATAP	
	Examiner Dhairya A. Patel	Art Unit 2151	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 4/10/2008.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-16 and 21-40 is/are pending in the application.
- 4a) Of the above claim(s) 17-20 is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-16, 21-40 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is responsive to communication filed on 4/10/2008. Claims 1-16,21-40 are subject to examination.
2. This amendment has been fully considered and entered.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 1-2,4-16, 28-38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kojima et al. U.S. patent Publication # 2003/0179705 (hereinafter Kojima) in view of Kaniz et al. U.S. Patent # 6,990,102 (hereinafter Kaniz) further in view of Hussain et al. U.S. Patent Publication # 2003/0223361 (hereinafter Hussain).

As per claim 1, Kojima teaches a method comprising:

-calculating traffic statistics associated with packet flows through a network (Paragraphs 46-48); **NOTE:** Kojima teaches traffic counter which stores and manages a traffic counter corresponding to the packet flow in the network. The input and output traffic counter has a function of receiving a packet, transmitting a packet, counting the amount of data (number of bytes) in packet in a network.

-storing the traffic statistics within a flow table (Fig. 8,9)(Paragraphs 61-64);

NOTE: The reference teaches storing in the table traffic counter collection and

managing the transmission and reception counts of each circuit based on the collection result of the traffic counter.

-maintaining a heap (Fig. 4 element "traffic count table") that provides a heap-ordered representation (Paragraph 56) of at least a portion of the traffic statistics of the packet flows within the flow table (Fig. 8,9)(Paragraphs 61-64),

Kojima teaches maintaining a traffic count table (i.e. heap because is a collection of data or table of data), and the traffic count table provides representation of packet flows i.e. transmission count and reception count

-processing the heap to select one or more of the packet flows (Paragraph 59-63); and

NOTE: Kojima teaches processing the initial table count table (Fig. 6) to select packet flows of A and B.

-outputting the traffic statistics associated with the selected packet flows (Paragraph 59-63)(Fig. 11)(Paragraphs 66-68). **NOTE:** Kojima teaches outputting the traffic count and the traffic count table stores the reception count and the transmission count w/ respect to packet flow A-B w/ respect to input circuit of A and output circuit of B (outputting traffic statistics associated w/ the selected packet flows).

Kojima is silent in teaches providing heap-ordered representation.

Kaniz teaches maintaining a heap that provides a heap-ordered representation of the packet flows (Fig. 6 element 604 "Heap entries")(column 9 lines 59-67)(column 10 lines 1-19). It would have been obvious to one of ordinary skill in the art at the time of applicant's invention was made to implement Kaniz's teaching in Kojima's teaching to

provide heap-ordered representation of the packet flows. The motivation for doing so is because heap-ordered representation is easier to search in table since heap-ordered tree is an array representation in which each node has a value which is less than value of parents which will decrease the time required to search the entire chain/tree.

Kojima and Kaniz are silent in teaching wherein the heap is a tree having a root node and a plurality of other nodes arranged as parent nodes and child nodes, each of the nodes storing traffic statistics for a different one of the packet flows, and wherein the tree is heap-ordered in that a value for the traffic statistics stored in any child node of the tree is less than or equal to a value for the traffic statistics stored within that child node's parent node within the tree.

Hussain teaches teach the heap is a tree (Fig. 3) having a root node (Fig. 3 element 306) and a plurality of other nodes (Fig. 3 element 302,304,305) arranged as parent nodes (Fig. 3 element 306) and child nodes, each of the nodes storing traffic statistics for a different one of the packet flows (Fig. 3 element 302 "packet flow A", 304 "packet flow B"), and wherein the tree is heap-ordered in that a value for the traffic statistics stored in any child node of the tree is less than or equal to a value for the traffic statistics stored within that child node's parent node within the tree (Paragraphs 47-49) and also heap-ordered representation (Fig. 3 element 302 "packet flow A", 304 "packet flow B") (Fig. 3 element 306). **NOTE:** The reference teaches that higher level node (Fig. 3 element 306) which is the parent node or the root node has defined for group of packet flows such as packet flows A and B, and the child nodes (Fig. 3 element 302,304) has each individual packet flow A (Fig. 3 element 302) and packet flow B (Fig.

3 element 304). This means that the data/traffic or packets for flows A and B are included in root node (Fig. 3 element 306) and child nodes are individual data or packets for packet A or B. This means that traffic or data for packet flows in the first level (Fig. 3 element 302 and 304) are less than or equal to the traffic or data for packet flows in the higher level (Fig. 3 element 306). It would have been obvious to one of ordinary skill in the art at the time of applicant's invention was made to implement Hussain's teaching in Kojima and Kaniz's teaching to come up with having a tree including root node and child node for traffic statistics in which parent node has more than or equal traffic statistics than child nodes. The motivation for doing so would be to arrange the data of the traffic statistics in the maximum value in a proper heap data structure, therefore heap data structure can be used as scheduling processes or generalized processor scheduling to ensure that each incoming data stream receives a certain portion of outgoing data stream bandwidth.

As per claim 2, Kojima, Kaniz and Hussain teaches the method of claim 1, but Kojima further teaches wherein processing the heap to select one or more of the packet flows comprises: receiving a query from a client (Paragraph 64); processing the heap in response to the query (Paragraph 66) to select one or more nodes of the tree while continuing to update the traffic flow statistics in the flow table in response to the network traffic; and outputting the traffic statistics associated with the selected packets to the clients (Paragraph 64-68)

As per claim 4, Kojima, Kaniz and Hussain teaches the method of claim 1, but Kojima further teaches wherein calculating traffic statistics and maintaining a heap are

performed concurrently with processing the heap and outputting the traffic statistics (Paragraph 64-68).

As per claim 5, Kojima, Kaniz and Hussain teaches the method of claim 1, but Kojima further teaches wherein maintaining a heap comprises maintaining the heap in real-time to reflect changes to the traffic statistics within flow table (Fig. 4,6,9,11)(Paragraph 63,66)

As per claim 6, Kojima, Kaniz and Hussain teaches the method of claim 1, but Kaniz further teaches wherein maintaining a table representation of the heap (Fig. 6 element 604 "Heap entries")(column 9 lines 59-67)(column 10 lines 1-19). It would have been obvious to one of ordinary skill in the art at the time of applicant's invention was made to implement Kaniz's teaching in Kojima's teaching to maintain a table representation of a heap. The motivation for doing so is because heap-ordered representation is easier to search in table since heap-ordered tree is an array representation in which each node has a value which is less than value of parents which will decrease the time required to search the entire chain/tree.

As per claim 7, Kojima, Kaniz and Hussain teaches the method of claim 6, but Kaniz further teaches wherein maintaining the heap comprises storing packet flow identifiers as elements of the table and in a heap-ordered fashion (column 8 lines 30-65).

As per claim 8, Kojima and Kaniz teaches the method of claim 1, but Kojima further teaches further comprising identifying the packet flows by at least one of a

source network address, a destination network address, a protocol, a source port number and a destination port number (Paragraph 49).

As per claim 9, Kojima and Kaniz teaches the method of claim 1, but Kaniz further teaches wherein calculating traffic statistics comprises: receiving a packet from a network via an interface card of a network device (column 5 lines 9-15); and extracting a key from a packet (column 5 lines 12-15); associating the packet with one of the packet flows based on the key (column 6 lines 1-26) ; and updating the flow statistics for the associated packet flow based on content of the packets (column 5 lines 41-67)(column 6 lines 1-6). It would have been obvious to one of ordinary skill in the art at the time of applicant's invention was made to implement Kaniz's teaching in Kojima's teaching to receiving packet, extract key from the packet, associate packet based on the key and update the flow statistics. The motivation for doing so would be to determine which port the packet/frame should be outputted i.e. using single port, multiple ports or all ports or no port based on the key (column 6 lines 18-20).

As per claim 10, Kojima and Kaniz teaches the method of claim 9, but Kaniz further teaches wherein maintaining a heap comprises performing a heapify operation on the heap after receiving the packet and updating the flow statistics (column 8 lines 30-39)(Fig. 6 "heap entries").

As per claim 11, Kojima and Kaniz teaches the method of claim 1, but Kojima further teaches wherein calculating traffic statistics further comprises communicating the packets to an accounting service card of a network device (Paragraph 47), wherein the accounting service card calculates the traffic statistics (Paragraph 46-47).

As per claim 12, Kojima and Kaniz teaches the method of claim 1, but Kaniz further teaches further comprising: receiving a query from a network client, wherein the query requests M ordered packet flows; cloning the heap to produce a heap clone in response to the query (column 10 lines 11-20)(Fig. 8 element "table 1 and table 2); performing M-1 heapify operations to extract the M ordered packet flow identifiers from the heap (column 8 lines 30-39)(column 5 lines 12-15); and outputting the traffic statistics associated with the M packet flow identifiers (column 5 lines 41-67)(column 6 lines 1-6). It would have been obvious to one of ordinary skill in the art at the time of applicant's invention was made to implement Kaniz's teaching in Kojima's teaching to receiving query regarding M ordered packet flows, cloning the heap and performing M-1 heapify operations. The motivation for doing so decrease the time required to search an entire chain/tree since the tree is heap-ordered tree.

As per claim 13, Kojima teaches the method comprising:

-maintaining a heap that provides a heap-ordered representation of packet flows within a network based on at least one criterion associated with the packet flows (Paragraph 56);

Kojima teaches maintaining a traffic count table (i.e. heap because is a collection of data or table of data), and the traffic count table provides representation of packet flows i.e. transmission count and reception count based on counting of amount of data (number of bytes) in a packet (one criteria associated w/ packet flows).

-processing the heap to output traffic statistics associated with an ordered subset of the packet flows. (Paragraph 59-63)(Fig. 11)(Paragraphs 66-68).

Kojima teaches processing the initial table count table (Fig. 6)(i.e. heap) to select packet flows of A and B and outputting the traffic count and the traffic count table stores the reception count and the transmission count w/ respect to packet flow A-B w/ respect to input circuit of A and output circuit of B (outputting traffic statistics associated w/ the packet flows).

Kojima is silent in teaches providing heap-ordered representation.

Kaniz teaches maintaining a heap that provides a heap-ordered representation of the packet flows (Fig. 6 element 604 "Heap entries")(column 9 lines 59-67)(column 10 lines 1-19). It would have been obvious to one of ordinary skill in the art at the time of applicant's invention was made to implement Kaniz's teaching in Kojima's teaching to provide heap-ordered representation of the packet flows. The motivation for doing so is because heap-ordered representation is easier to search in table since heap-ordered tree is an array representation in which each node has a value which is less than value of parents.

Kojima and Kaniz are silent in teaching wherein the heap is a tree having a root node and a plurality of other nodes arranged as parent nodes and child nodes, each of the nodes storing traffic statistics for a different one of the packet flows, and wherein the tree is heap-ordered in that a value for the traffic statistics stored in any child node of the tree is less than or equal to a value for the traffic statistics stored within that child node's parent node within the tree.

Hussain teaches wherein the heap is a tree (Fig. 3) having a root node (Fig. 3 element 306) and a plurality of other nodes (Fig. 3 element 302,304,305) arranged as

parent nodes (Fig. 3 element 306) and child nodes, each of the nodes storing traffic statistics for a different one of the packet flows (Fig. 3 element 302 “packet flow A”, 304 “packet flow B”), and wherein the tree is heap-ordered in that a value for the traffic statistics stored in any child node of the tree is less than or equal to a value for the traffic statistics stored within that child node’s parent node within the tree (Paragraphs 47-49) **NOTE:** The reference teaches that higher level node (Fig. 3 element 306) which is the parent node or the root node has defined for group of packet flows such as packet flows A and B, and the child nodes (Fig. 3 element 302,304) has each individual packet flow A (Fig. 3 element 302) and packet flow B (Fig. 3 element 304). This means that the data/traffic or packets for flows A and B are included in root node (Fig. 3 element 306) and child nodes are individual data or packets for packet A or B. This means that traffic or data for packet flows in the first level (Fig. 3 element 302 and 304) are less than or equal to the traffic or data for packet flows in the higher level (Fig. 3 element 306). It would have been obvious to one of ordinary skill in the art at the time of applicant’s invention was made to implement Hussain’s teaching in Kojima and Kaniz’s teaching to come up with having a tree including root node and child node for traffic statistics in which parent node has more than or equal traffic statistics than child nodes. The motivation for doing so would be to arrange the data of the traffic statistics in the maximum value in a proper heap data structure, therefore heap data structure can be used as scheduling processes or generalized processor scheduling to ensure that each incoming data stream receives a certain portion of outgoing data stream bandwidth.

As per claim 14, Kojima and Kaniz teaches the method of claim 13, but Kojima further teaches the criteria comprises one of a byte count and a packet count associated w/ each of the packet flows (Paragraph 46)(Fig. 4,6,9,11 "Flow table A-B")

As per claim 15, Kojima and Kaniz teaches the method of claim 13, but Kaniz further teaches receiving a query from a network client, cloning the heap to produce a heap clone (column 10 lines 11-20)(Fig. 8 element "table 1 and table 2"); extracting one or more heap entries from a heap clone to extract identifiers for the ordered subset of the packet flows (column 8 lines 30-39)(column 5 lines 12-15); and outputting the traffic statistics associated with the extracted identifiers (column 5 lines 41-67)(column 6 lines 1-6). It would have been obvious to one of ordinary skill in the art at the time of applicant's invention was made to implement Kaniz's teaching in Kojima's teaching to receiving query regarding M ordered packet flows, cloning the heap and extracting heap entries and outputting traffic statistics. The motivation for doing so decrease the time required to search an entire chain/tree since the tree is heap-ordered tree.

As per claim 16, Kojima and Kaniz teaches the method of claim 13, but Kojima further teaches wherein maintaining a heap is performed concurrently with processing the heap (Paragraph 64-68).

As per claim 28, Kojima teaches a network elements comprising:

- an interface (Fig. 1 element 104) to receive packets flows from a network(Paragraphs 46-48);

- a control unit (Fig. 1 element 101 "control unit")coupled to the interface, wherein the control unit computes flow statistics for the packets, (Paragraphs 46-48);

Kojima teaches traffic counter which stores and manages a traffic counter corresponding to the packet flow in the network. The input and output traffic counter has a function of receiving a packet, transmitting a packet, counting the amount of data (number of bytes) in packet in a network.

Kojima is silent in teaching maintaining identifiers for the packet flows in a heap-ordered representation based on at least one of the statistics. Kaniz teaches maintaining identifiers for the packet flows in a heap-ordered representation based on at least one of the statistics (column 8 lines 30-65). It would have been obvious to one of ordinary skill in the art at the time of applicant's invention was made to implement Kaniz's teaching in Kojima's teaching to maintain identifier in a heap-ordered representation of the packet flows. The motivation for doing so is because heap-ordered representation is easier to search in table since heap-ordered tree is an array representation in which each node has a value which is less than value of parents which will decrease the time required to search the entire chain/tree.

Kojima and Kaniz are silent in teaching wherein the heap is a tree having a root node and a plurality of other nodes arranged as parent nodes and child nodes, each of the nodes storing traffic statistics for a different one of the packet flows, and wherein the tree is heap-ordered in that a value for the traffic statistics stored in any child node of the tree is less than or equal to a value for the traffic statistics stored within that child node's parent node within the tree.

Hussain teaches wherein the heap is a tree (Fig. 3) having a root node (Fig. 3 element 306) and a plurality of other nodes (Fig. 3 element 302,304,305) arranged as

parent nodes (Fig. 3 element 306) and child nodes, each of the nodes storing traffic statistics for a different one of the packet flows (Fig. 3 element 302 “packet flow A”, 304 “packet flow B”), and wherein the tree is heap-ordered in that a value for the traffic statistics stored in any child node of the tree is less than or equal to a value for the traffic statistics stored within that child node’s parent node within the tree (Paragraphs 47-49) **NOTE:** The reference teaches that higher level node (Fig. 3 element 306) which is the parent node or the root node has defined for group of packet flows such as packet flows A and B, and the child nodes (Fig. 3 element 302,304) has each individual packet flow A (Fig. 3 element 302) and packet flow B (Fig. 3 element 304). This means that the data/traffic or packets for flows A and B are included in root node (Fig. 3 element 306) and child nodes are individual data or packets for packet A or B. This means that traffic or data for packet flows in the first level (Fig. 3 element 302 and 304) are less than or equal to the traffic or data for packet flows in the higher level (Fig. 3 element 306). It would have been obvious to one of ordinary skill in the art at the time of applicant’s invention was made to implement Hussain’s teaching in Kojima and Kaniz’s teaching to come up with having a tree including root node and child node for traffic statistics in which parent node has more than or equal traffic statistics than child nodes. The motivation for doing so would be to arrange the data of the traffic statistics in the maximum value in a proper heap data structure, therefore heap data structure can be used as scheduling processes or generalized processor scheduling to ensure that each incoming data stream receives a certain portion of outgoing data stream bandwidth.

As per claim 29, Kojima and Kaniz teaches the network elements of claim 28, but Kaniz further teaches wherein the control unit maintains identifiers as a heap having plurality of entries, wherein each entry stores a respective identifier for one of the packet flows (Fig. 6 element 604 "Heap entries")(column 9 lines 59-67)(column 10 lines 1-19).

As per claim 30, Kojima and Kaniz teaches the network elements of claim 29, further comprising a user interface to receive a query from client (Paragraph 64). Kojima is silent in teaching wherein the control unit processes the heap in response to the query to select one or more identifiers, and the user interface outputs a portion of the traffic statistics associated with the selected identifiers. Kaniz teaches the control unit processes the heap in response to the query to select one or more identifiers, and the user interface outputs a portion of the traffic statistics associated with the selected identifiers (column 10 lines 1-35). It would have been obvious to one of ordinary skill in the art at the time of applicant's invention was made to implement Kaniz's teaching in Kojima's teaching to process the heap in response to the query and user interface outputs a portion of the traffic statistics. The motivation for doing so would be to keep up-to-date on the network traffic statistics associated with each packet flows and its identifiers.

As per claim 31, Kojima and Kaniz teaches the network elements of claim 29, but Kaniz further teaches wherein the control unit clones the heap to produce a heap clone, and processes the heap clone to extracting the identifiers (column 8 lines 30-39)(column 5 lines 12-15). It would have been obvious to one of ordinary skill in the art at the time of applicant's invention was made to implement Kaniz's teaching in Kojima's

teaching to receiving query regarding M ordered packet flows, cloning the heap and extracting heap entries. The motivation for doing so decrease the time required to search an entire chain/tree since the tree is heap-ordered tree

As per claim 32, Kojima and Kaniz teaches the network elements of claim 29, but Kojima further teaches wherein the control unit concurrently computes the flow statistics and maintains the identifiers concurrently with processing the heap and outputting the portion of the traffic statistics (Paragraph 64-68).

As per claim 33, Kojima and Kaniz teaches the network elements of claim 29, but Kojima further teaches wherein the control unit associates the packets with the packet flows by determining for each packet at least one of a source network address, a destination network address, a protocol, a source port number, and a destination port number (Paragraph 49).

As per claim 34, Kojima and Kaniz teaches the network elements of claim 28, Kojima further teaches wherein the control unit calculates the flow statistics (Fig. 4,6,9,11)(Paragraph 63,66) and maintains the heap-ordered representation in real-time as packets are received from the interface card (Paragraph 63-68).

Kojima does not explicitly teaches maintaining the heap-ordered representation. Kaniz teaches maintaining a heap that provides a heap-ordered representation (Fig. 6 element 604 "Heap entries")(column 9 lines 59-67)(column 10 lines 1-19). It would have been obvious to one of ordinary skill in the art at the time of applicant's invention was made to implement Kaniz's teaching in Kojima's teaching to provide heap-ordered representation of the packet. The motivation for doing so is because heap-ordered representation is

easier to search in table since heap-ordered tree is an array representation in which each node has a value which is less than value of parents which will decrease the time required to search the entire chain/tree

As per claim 35, Kojima and Kaniz teaches the network elements of claim 28, wherein the control unit comprises an accounting module that calculates the traffic statistics (Fig. 1 element 121, 123,122)

As per claim 36, Kojima and Kaniz teaches the network elements of claim 28, but Kojima further teaches further comprising an accounting service card (Fig. 1 element 122, 123), wherein the control unit forwards the packets to the accounting service card to calculate the traffic statistics (Paragraph 47-48).

As per claim 37, Kojima and Kaniz teaches the network elements of claim 28, but Kojima further teaches wherein the control unit comprises: a routing engine to maintain routing information representing a topology of the network (Paragraph 8); and a forwarding engine to forward the network packets in accordance with the routing information (Paragraph 54).

As per claim 38, Kojima and Kaniz teaches the network elements of claim 28, but Kojima further teaches wherein the network device comprises one of a router, a switch, a gateway and a hub (Fig. 1 element 102).

Claim 3 is rejected under 35 U.S.C. 103(a) as being unpatentable over Kojima et al. U.S. patent Publication # 2003/0179705 (hereinafter Kojima) in view of Kaniz et al. U.S. Patent # 6,990,102 (hereinafter Kaniz) further in view of

Hussain et al. U.S. Patent Publication # 2003/0223361 (hereinafter Hussain) further in view of Hirono et al. U.S. Patent # 6,910,213 (hereinafter Hirono)

As per claim 3, Kojima, Kaniz and Hussain teaches the method of claim 1, but are silent in teaching wherein processing the heap comprises: cloning the heap to produce a heap clone in the form of a tree having a root node a plurality of other nodes arranged as parent nodes and child nodes, the nodes of heap clone storing copies of the traffic statistics stored by the nodes of the heap and extracting one or more nodes from the heap clone to select the packet flows. Hirono teaches cloning the heap to produce a heap clone in the form of a tree having a root node a plurality of other nodes arranged as parent nodes and child nodes, the nodes of heap clone storing copies of the traffic statistics stored by the nodes of the heap (column 22 lines 40-56)(Fig. 20); and extracting one or more nodes from the heap clone to select the packet flows (column 23 lines 19-25)(Fig. 20). It would have been obvious to one of ordinary skill in the art at the time of applicant's invention was made to implement Hirono's teaching in Kojima, Kaniz and Hussain's teaching to cloning the heap in form of a tree having a root node and plurality of other nodes. The motivation for doing so would be to having a backup data, therefore a clone heap is created so if the original heap area/memory/sector gets corrupted or erased, one can use the clone/copied heap to retrieve original data.

Claims 21-28,39-40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kojima et al. U.S. patent Publication # 2003/0179705

(hereinafter Kojima) in view of Kaniz et al. U.S. Patent # 6,990,102 (hereinafter Kaniz)

As per claim 21, Kojima teaches a computer-readable medium comprising instructions for causing a programmable processor to:

- calculate traffic statistics associated with packet flows through a network (Paragraphs 46-48);

Kojima teaches traffic counter which stores and manages a traffic counter corresponding to the packet flow in the network. The input and output traffic counter has a function of receiving a packet, transmitting a packet, counting the amount of data (number of bytes) in packet in a network.

- maintain a heap (Fig. 4 element "traffic count table") that provides a heap-ordered representation of the packet flows (Paragraph 56);

Kojima teaches maintaining a traffic count table (i.e. heap because is a collection of data or table of data), and the traffic count table provides representation of packet flows i.e. transmission count and reception count

- process the heap to select one or more of the packet flows (Paragraph 59-63); and

Kojima teaches processing the initial table count table (Fig. 6) to select packet flows of A and B.

- output the traffic statistics associated with the selected packet flows (Paragraph 59-63)(Fig. 11)(Paragraphs 66-68).

Kojima teaches outputting the traffic count and the traffic count table stores the reception count and the transmission count w/ respect to packet flow A-B w/ respect to input circuit of A and output circuit of B (outputting traffic statistics associated w/ the selected packet flows).

Kojima is silent in teaches providing heap-ordered representation.

Kaniz teaches maintaining a heap that provides a heap-ordered representation of the packet flows (Fig. 6 element 604 "Heap entries")(column 9 lines 59-67)(column 10 lines 1-19). It would have been obvious to one of ordinary skill in the art at the time of applicant's invention was made to implement Kaniz's teaching in Kojima's teaching to provide heap-ordered representation of the packet flows. The motivation for doing so is because heap-ordered representation is easier to search in table since heap-ordered tree is an array representation in which each node has a value which is less than value of parents which will decrease the time required to search the entire chain/tree.

As per claim 22, Kojima and Kaniz teaches the computer-readable medium of claim 21, but Kojima further teaches further comprising instructions to cause the processor to: receiving a query from a client (Paragraph 64); processing the heap in response to the query (Paragraph 66); and outputting the traffic statistics associated with the selected packets to the clients (Paragraph 64-68)

As per claim 23, Kojima and Kaniz teaches the computer-readable medium of claim 21, but Kaniz further teaches further comprising instructions to cause the processor to: cloning the heap to produce a heap clone (column 10 lines 11-20)(Fig. 8 element "table 1 and table 2); and extracting one or more heap entries from the heap

clone to select the packet flows (column 8 lines 30-39)(column 5 lines 12-15). It would have been obvious to one of ordinary skill in the art at the time of applicant's invention was made to implement Kaniz's teaching in Kojima's teaching to receiving query regarding M ordered packet flows, cloning the heap and extracting heap entries. The motivation for doing so decrease the time required to search an entire chain/tree since the tree is heap-ordered tree.

As per claim 24, Kojima and Kaniz teaches the computer-readable medium of claim 21, but Kojima further teaches further comprising instructions to cause the processor to perform calculating traffic statistics and maintaining a heap are performed concurrently with processing the heap and outputting the traffic statistics (Paragraph 64-68).

As per claim 25, Kojima and Kaniz teaches the computer-readable medium of claim 21, but Kojima further teaches further comprising instructions to cause the processor to maintain a flow that stores traffic flow statistics for a set of packet flows (Fig. 4,6,9,11)(Paragraph 63,66)

As per claim 26, Kojima and Kaniz teaches the computer-readable medium of claim 21, but Kaniz further teaches wherein maintaining a table representation of the heap (Fig. 6 element 604 "Heap entries")(column 9 lines 59-67)(column 10 lines 1-19). It would have been obvious to one of ordinary skill in the art at the time of applicant's invention was made to implement Kaniz's teaching in Kojima's teaching to maintain a table representation of a heap. The motivation for doing so is because heap-ordered representation is easier to search in table since heap-ordered tree is an array

representation in which each node has a value which is less than value of parents which will decrease the time required to search the entire chain/tree.

As per claim 27, Kojima and Kaniz teaches the computer-readable medium of claim 21, but Kaniz further teaches further comprising instructions to cause the processor to store packet flow identifiers as elements of the table and in a heap-ordered fashion (column 8 lines 30-65)

As per claim 39, Kojima teaches a network device comprising: an interface (Fig. 1 element 104) to receive packet flows from a network; a control unit (Fig. 1 element "control unit") coupled to the interface (Paragraphs 46-48); and an accounting service card (Fig. 1 element 122,123), wherein the control unit forwards the packets to the accounting service card to calculate the traffic statistics, wherein the accounting service card computes flow statistics for the packets (Paragraph 47-48).

Kojima is silent in teaches maintaining a heap that provides a heap-ordered representation of the packet flows. Kaniz teaches maintaining a heap that provides a heap-ordered representation of the packet flows (Fig. 6 element 604 "Heap entries")(column 9 lines 59-67)(column 10 lines 1-19). It would have been obvious to one of ordinary skill in the art at the time of applicant's invention was made to implement Kaniz's teaching in Kojima's teaching to provide heap-ordered representation of the packet flows. The motivation for doing so is because heap-ordered representation is easier to search in table since heap-ordered tree is an array representation in which each node has a value which is less than value of parents which will decrease the time required to search the entire chain/tree.

As per claim 40, Kojima and Kaniz teaches the network device of claim 39, but Kojima further teaches wherein the accounting service card comprises: a flow table to store the flow statistics (Fig. 6,9, 11) but fails to teach heap table to store the heap. Kaniz teaches a heap table to store the heap (Fig. 6 element 604 "Heap entries")(column 9 lines 59-67)(column 10 lines 1-19) It would have been obvious to one of ordinary skill in the art at the time of applicant's invention was made to implement Kaniz's teaching in Kojima's teaching to provide a heap table to store the heap. The motivation for doing so is because a heap table is easier to search in table since heap-ordered tree is an array representation in which each node has a value which is less than value of parents which will decrease the time required to search the entire chain/tree.

Response to Arguments

Applicant's arguments filed 4/10/2008 have been fully considered but they are not persuasive.

A). Applicant states Kojima and Kaniz fails to teach "calculating traffic statistics associated with packet flows through a network; storing the traffic statistics within a flow table; maintaining a heap that provides a heap-ordered representation". Examiner respectfully disagrees with the applicant because in Paragraphs 46-48 Kojima teaches calculating traffic statistics associated with packet flows through a network, particularly traffic counter which stores and manages a traffic counter corresponding to the packet flow in the network. The input and output traffic counter has a function of receiving a packet, transmitting a packet, counting the amount of data (number of bytes) in packet

in a network. The traffic statistics are considered as receiving a packet, transmitting a packet, counting the amount of data (number of bytes) in packet in a network.

In Fig. 8,9, Paragraphs 61-64, Kojima teaches storing the traffic statistics within a flow table. Kojima particularly states storing in the table traffic counter collection and managing the transmission and reception counts of each circuit based on the collection result of the traffic counter. Kojima also teaches maintaining a heap (Fig. 4 element "traffic count table") that provides a heap-ordered representation (Paragraph 56) of at least a portion of the traffic statistics of the packet flows within the flow table (Fig. 8,9)(Paragraphs 61-64), Kojima teaches maintaining a traffic count table (i.e. heap because is a collection of data or table of data), and the traffic count table provides representation of packet flows i.e. transmission count and reception count.

Applicants also argue that Kaniz does not teach a heap-ordered representation. Kaniz teaches having a having heap entries and having addresses from "n" to "4095". Kaniz and Kojima does not particularly state that wherein the heap is a tree having a root node and a plurality of other nodes arranged as parent nodes and child nodes, each of the nodes storing traffic statistics for a different one of the packet flows, and wherein the tree is heap-ordered in that a value for the traffic statistics stored in any child node of the tree is less than or equal to a value for the traffic statistics stored within that child node's parent node within the tree.

Hussain teaches teach the heap is a tree (Fig. 3) having a root node (Fig. 3 element 306) and a plurality of other nodes (Fig. 3 element 302,304,305) arranged as parent nodes (Fig. 3 element 306) and child nodes, each of the nodes storing traffic

statistics for a different one of the packet flows (Fig. 3 element 302 “packet flow A”, 304 “packet flow B”), and wherein the tree is heap-ordered in that a value for the traffic statistics stored in any child node of the tree is less than or equal to a value for the traffic statistics stored within that child node’s parent node within the tree (Paragraphs 47-49) and also heap-ordered representation (Fig. 3 element 302 “packet flow A”, 304 “packet flow B”) (Fig. 3 element 306). **NOTE:** The reference teaches that higher level node (Fig. 3 element 306) which is the parent node or the root node has defined for group of packet flows such as packet flows A and B, and the child nodes (Fig. 3 element 302,304) has each individual packet flow A (Fig. 3 element 302) and packet flow B (Fig. 3 element 304). This means that the data/traffic or packets for flows A and B are included in root node (Fig. 3 element 306) and child nodes are individual data or packets for packet A or B. This means that traffic or data for packet flows in the first level (Fig. 3 element 302 and 304) are less than or equal to the traffic or data for packet flows in the higher level (Fig. 3 element 306). This provides the functionality of the heap and heap-ordered representation. It would have been obvious to one of ordinary skill in the art at the time of applicant’s invention was made to implement Hussain’s teaching in Kojima and Kaniz’s teaching to come up with having a tree including root node and child node for traffic statistics in which parent node has more than or equal traffic statistics than child nodes. The motivation for doing so would be to arrange the data of the traffic statistics in the maximum value in a proper heap data structure, therefore heap data structure can be used as scheduling processes or generalized processor scheduling to

ensure that each incoming data stream receives a certain portion of outgoing data stream bandwidth.

Conclusion

3. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.
4. A). "Packet Transfer apparatus with the function of flow detection and flow management method" by Yazaki et al a. U.S. patent # 7,031,313
5. A shortened statutory period for response to this action is set to expire **3 (three) months and 0 (zero) days** from the mail date of this letter. Failure to respond within the period for response will result in **ABANDONMENT** of the applicant (see 35 U.S.C 133, M.P.E.P 710.02, 710.02(b)).

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

6.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dhairya A. Patel whose telephone number is 571-272-5809. The examiner can normally be reached on Monday-Friday 7:00AM-4:30PM, first Fridays OFF.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Follansbee can be reached on 571-272-3964. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Application/Control Number: 10/670,944

Page 27

Art Unit: 2151

DAP

/John Follansbee/

Supervisory Patent Examiner, Art Unit 2151